WEST

Freeform Search

Database: US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: (modulo or modulus) same (binary or bit) same dividend same divisor same (remainder or residue)

Display: 20 **Documents in Display Format:** CIT **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

Search History

DATE: Friday, March 01, 2002 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L10	('6275311')[ABPN1,NRPN,PN,TBAN,WKU]	2	L10
L9	('EP 444661A'| 'SU 1772802A'| '4759063'| '6275311'| '4839839'| '6175850'| 'NN74033323'| 'JP410187037A'| '3916095'| '5396595'| 'JP410307710A')[ABPN1,NRPN,PN,TBAN,WKU]	23	L9
L8	(modulo or modulus) same (binary or bit) same dividend same divisor same (remainder or residue)	11	L8
L7	(modulo or modulus) same (binary or bit) same dividend same divisor same (remainder or residue) same (indirect$4 or direct$4 or non$1recur$7 or recur$7 or non$1iterat$7 or iterat$7)	1	L7
L6	(modulo or modulus) same (binary or bit) same dividend same divisor same (remainder or residue) same (indirect$4 or direct$4 or non$1recur$7 or recur$7 or non$1iterat$7 or iterat$7)	1	L6
L5	('EP 444661A')[ABPN1,NRPN,PN,TBAN,WKU]	1	L5
L4	l3 and (modulo or modulus) same (binary or bit) same dividend same divisor same remainder same (indirect$4 or direct$4 or non$1recur$7 or recur$7 or non$1iterat$7 or iterat$7)	1	L4
L3	l2 and (binary or bit) same dividend same divisor same remainder same (indirect$4 or direct$4 or non$1recur$7 or recur$7 or non$1iterat$7 or iterat$7)	124	L3
L2	l1 and dividend same divisor same remainder same (indirect$4 or direct$4 or non$1recur$7 or recur$7 or non$1iterat$7 or iterat$7)	173	L2
L1	dividend same divisor same remainder	732	L1

END OF SEARCH HISTORY

WEST

Freeform Search

Database: US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: l2 and (modulo or modulus or moduli)

Display: 20 **Documents in Display Format:** CIT **Starting with Number** 1

Generate: ○ Hit List ◉ Hit Count ○ Side by Side ○ Image

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences | Cases

Search History

DATE: Friday, March 01, 2002 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ			
L4	('5426600')[ABPN1,NRPN,PN,TBAN,WKU]	3	L4
L3	l2 and (modulo or modulus or moduli)	1	L3
L2	l1 and "n-bit" divisor and (remainder or residue)	17	L2
L1	"n-bit" divisor	19	L1

END OF SEARCH HISTORY

data block, whilst at the subsequent transitions the (i)th flip-flop accepts the signals present at the first input, the first or second inputs being enabled according to the logic level of a signal (L) active in correspondence with the first bit of a data block.

WEST

End of Result Set

Generate Collection Print

L8: Entry 11 of 11 File: DWPI May 19, 1998

DERWENT-ACC-NO: 1991-261439
DERWENT-WEEK: 199831

TITLE: Digital signal error correction code generator using remainder - obtd. in polynomial division with dividend polynomial coeffts. defined by bits of each serial data block

Equivalent Abstract Text:
An electronic circuit for generating an error detection code for errors in digital signals (D), organized in data blocks forming the coefficients of a dividend polynomial, to be divided by a suitable divisor polynomial of (n)th degree to obtain a remainder polynomial of (n-1)th degree, whose coefficients (R1,R2,R3,R4) form the error detection code, the circuit consisting of a shift register composed of n flip-flops (FF1,...,FF4) wherein the signal outgoing from the (i-1)th flip-flop (FF1), i being comprised between 0 and n, is sent to a first input of (i)th flip-flop (FF2) directly or through a modulo-2-adder (EO1) according to whether the coefficient to the term of (i)the degree of th divisor polynomial is equal to or different from zero, said adder (EO1) adding to the signal outgoing from the preceding flip-flop (FF1) a feedback signal (10) obtained at the output of the (n)th flip-flop (FF4) which is the MSB flip-flop, either directly, or after adding it to the input data signal (D) in a modulo-2-adder (EO2) which for providing the feedback signal (10) receives as inputs the MSB of the remainder (R4) and said input data signal (D), the feedback signal (10) being supplied to the (i-1) th flip-flop (FF1) and to the other modulo-2 adder (EO1), and all the flip-flops being clocked by a common clock signal (C), characterized in that a second input of the (i)t flip-flop (FF2) accepts the first bit of said data block or a bit at low logic value, according to whether the coefficient of the term of (i)th degree of th divisor polynomial to respectively different from or equal to zero, in correspondence with the clock signal (C) transition relevant to the first bit of the data block, whilst at the subsequent transitions the (i)th flip-flop accepts the signals present at the first input, the first or second inputs being enabled according to the logic level of a signal (L) active in correspondence with the first bit of a data block.

Equivalent Abstract Text (1):
An electronic circuit for generating an error detection code for errors in digital signals (D), organized in data blocks forming the coefficients of a dividend polynomial, to be divided by a suitable divisor polynomial of (n)th degree to obtain a remainder polynomial of (n-1)th degree, whose coefficients (R1,R2,R3,R4) form the error detection code, the circuit consisting of a shift register composed of n flip-flops (FF1,...,FF4) wherein the signal outgoing from the (i-1)th flip-flop (FF1), i being comprised between 0 and n, is sent to a first input of (i)th flip-flop (FF2) directly or through a modulo-2-adder (EO1) according to whether the coefficient to the term of (i)the degree of th divisor polynomial is equal to or different from zero, said adder (EO1) adding to the signal outgoing from the preceding flip-flop (FF1) a feedback signal (10) obtained at the output of the (n)th flip-flop (FF4) which is the MSB flip-flop, either directly, or after adding it to the input data signal (D) in a modulo-2-adder (EO2) which for providing the feedback signal (10) receives as inputs the MSB of the remainder (R4) and said input data signal (D), the feedback signal (10) being supplied to the (i-1) th flip-flop (FF1) and to the other modulo-2 adder (EO1), and all the flip-flops being clocked by a common clock signal (C), characterized in that a second input of the (i)t flip-flop (FF2) accepts the first bit of said data block or a bit at low logic value, according to whether the coefficient of the term of (i)th degree of th divisor polynomial to respectively different from or equal to zero, in correspondence with the clock signal (C) transition relevant to the first bit of the

L8: Entry 9 of 11 File: TDBD Mar 1, 1974

DOCUMENT-IDENTIFIER: NN74033323
TITLE: Residue Checking with Array Logic. March 1974.

Disclosure Text (1):
4p. In computers the primitive operations done are usually well checked by carrying along parity bits and predicting them where necessary. In high-speed units where quite complex functions are performed, often at cycle rate faster than the basic CPU this may not be possible without affecting performance or circuit count unacceptably. In such cases a residue system may be chosen. - The general method of finding a residue, is to divide the number by the modulus chosen in the arithmetic appropriate to the base of the number. The quotient is discarded and the remainder is the residue. If the modulus is one less than the base, then the residue of the sum of the digits is the residue of the number. Thus the familiar "casting out nines" of decimal arithmetic which becomes "casting out fifteens" in hexadecimal. To find a residue by division, the number must be available starting with the high order while the sum residue may be obtained in any order. On the other hand, 9 and 15 are not quite as desirable as prime numbers are for moduli. - Currently available array logic is well suited for generating and manipulating residues up to four bits in length, and its flexibility allows free choice of modulus and base within the bit limitation. Two examples will be given. Checking High-Speed Multiply (Fig. 1) In the IBM Technical Disclosure Bulletin December 1973, Vol. 16, No. 7, pages 2195 to 2198, a unit is shown, into which a number to be of the operation, the multiplier is fed into the unit digit-by-digit from a low order and after a delay product digits start becoming available, again from the low order. A modulus of 15 makes best use of these flows. A residue adder can be attached to each digit bus. Each adder would consist of a 256 x 4 array module and 4 bits of latching, so a partial residue could be reentered. The array would be programmed to add two digits modulo 15. At about the half-way point the multiplier register M Reg becomes idle and if the multiplicand (MC) is set into it, the same adder can accumulate the MC residue. - Concurrently, an identical unit on the output bus is obtaining the product residue. When both input factors have been processed the results are fed to another 256 x 4 array module, this one programmed to do modulo 15 multiplication. This output should agree with the actual product residue. Checking High-Speed Convert (Fig. 2) In the IBM Technical Disclosure Bulletin, December 1973, Vol. 16, No. 7, pages 2195 to 2198, a unit is shown, into which a number to be converted is inserted, a digit at a time, from the high end and then the result is extracted digit-by-digit starting with the low end. If the input string is divided by one less than the base of the converted output, this residue can be held for comparison with the sum of the output digits accumulated with the same modulus. - A prime modulus can also be used (see Fig. 3). While in general, divide must proceed from the high-order end, if a supposed remainder is available at the start and if the divisor is relatively prime to the arithmetic base, then the partial remainders can be obtained in reverse order. Another way of saying this is that the high-order dividend digit can be predicted from the low-order digit, the remainder and the fixed divisor. The final output of this process must be zero or an error has occurred in conversion.

L8: Entry 5 of 11 File: USPT Jul 19, 1988

DOCUMENT-IDENTIFIER: US 4759063 A
TITLE: Blind signature systems

Detailed Description Paragraph Right (37):
Referring now to FIG. 3, a detailed description of an exemplary embodiment of a modular multiplicative inverter, herein called a modular inverter, is presented. The number to be inverted appears on line 351, and is initially loaded into register 301. The output of register 301 appears on line 352, which is input to register 302, which takes its initial value, the modulus n, from line 353, and has output on line 354. Ordinary arithmetic divider 303 takes its dividend from line 354 and its divisor from line 352; its quotient output appears on line 355 and its remainder output appears on line 356. Such binary arithmetic dividers for unsigned integers are well known in the art, for example see K. Hwang, "Computer Arithmetic: principles, architecture, and design" John Wiley, 1979, Chapter 7. Line 356 is input to register 301, described earlier. Line 355 is input to modular multiplier 304, such multipliers to be described. The output of modular multiplier 304 is line 357, which is subtrahend input to modular subtractor 305, such subtractors to be described. The difference output of modular subtractor 305 is line 358, which is input to register 306, which takes its initial value of 1 from line 359 and whose output appears on line 360. Modular multiplier 304 already described takes one of its multiplicand inputs from line 360. Register 307 takes input from line 360, takes its initial value of 0 from line 361, and its output appears on line 362. Line 362 is minuend input for modular subtractor 305 already described, and is the output of the modular inverter.

L8: Entry 3 of 11 File: USPT Mar 7, 1995

DOCUMENT-IDENTIFIER: US 5396595 A
TITLE: Method and system for compression and decompression of data

Detailed Description Paragraph Right (7):
In the example of FIG. 3, the suffix length is 2. In a preferred embodiment, the suffix length can vary from data buffer to data buffer. The suffix length for a data buffer is selected to minimize the size of the resulting compressed data buffer as explained in detail below. The indices are encoded in modulo arithmetic format based on the suffix length. In modulo arithmetic format, a number is represented by the integer portion of a quotient, a remainder, and a base. For example, the number 51 is represented in modulo base 4 format as (12, 3), that is, an integer portion of 12 and remainder of 3 (12*4+3=51) and represented in modulo base 5 as (10, 1), that is, an integer portion of 10 and a remainder of 1 (10*5+1=51). In modulo arithmetic format, the base is the divisor and the number to be represented is the dividend. In the present invention, the index is the number to be represented (the dividend) and 2 to the power of the suffix length (2.sup.suffixlength) is the base (divisor). The prefix represents the quotient, and the suffix represents the remainder. The number 1s in the prefix is equal to the quotient. For example, if quotient is 3, then the prefix is "111." The remainder is stored in binary arithmetic format. For example, if the remainder is 5 and the suffix length is 4, then the suffix is "0101." The divisor is defined by the selection of the suffix length. The divisor is 2 to the power of the suffix length. For example, if the suffix length is 4 then the divisor is 16(2.sup.4).

WEST

Generate Collection Print

L8: Entry 2 of 11 File: USPT Jan 16, 2001

DOCUMENT-IDENTIFIER: US 6175850 B1
TITLE: Scheme for carrying out modular calculations based on redundant binary calculation

Abstract Paragraph Left (1):
A scheme for carrying out modular calculations which is capable of carrying out modular calculations using redundant binary calculation even when a number of bits of the mantissa (dividend) is larger than a number of bits of the modulus (divisor). In this scheme, the divisor c in the divisor register is left shifted by (i-j) digits when a number of digits j of the divisor c is less than a number of digits i that can be stored in the divisor register, and the modular reduction a mod c is calculated up to (i-j)-th decimal place using the dividend a and the left shifted divisor c. Alternatively, the divisor c given in h-ary notation in the divisor register is left shifted by (i-j) digits when a number of digits j of the divisor c is less than a number of digits i that can be stored in the divisor register, while the dividend a given in h-ary notation in the dividend register is left shifted by (k-l) digits when a number of digits l of the dividend a is less than a number of digits k that can be stored in the dividend register, where k.gtoreq.i. Then, the modular reduction a mod c is calculated up to a digit of [(k-l)-(i-j)]-th power of h using the left shifted dividend a and the left shifted divisor c to obtain a remainder, and this remainder is right shifted by (k-l) digits.

WEST

End of Result Set

Generate Collection Print

L4: Entry 1 of 1 File: DWPI May 19, 1998

DERWENT-ACC-NO: 1991-261439
DERWENT-WEEK: 199831

TITLE: Digital signal error correction code generator using remainder - obtd. in polynomial division with dividend polynomial coeffts. defined by bits of each serial data block

INVENTOR: GANDINI, M; GHIGO, G ; MARCHISIO, M

PATENT-ASSIGNEE:

ASSIGNEE	CODE
SIP SOC ITAL ESERCIZIO TELECOM	SIPIN
SIP SOC ITAL ESERCIZIO TELECOM	SIPIN
SIP SOC ITAL ESERCIZIO TELEFONICA	SIPIN

PRIORITY-DATA: 1990IT-0067146 (March 1, 1990)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
CA 2037219 C	May 19, 1998		000	H03M013/00
EP 444661 A	September 4, 1991		005	
CA 2037219 A	September 2, 1991		000	
JP 06045952 A	February 18, 1994		000	H03M013/00
US 5309449 A	May 3, 1994		005	G06F011/10
IT 1241429 B	January 17, 1994		000	H04L000/00
EP 444661 B1	July 10, 1996	E	006	H03M013/00
DE 69120698 E	August 14, 1996		000	H03M013/00

DESIGNATED-STATES: BE DE FR GB IT NL SE BE DE FR GB IT NL SE

CITED-DOCUMENTS:A3...9150; EP 225761 ; NoSR.Pub ; US 4723244 ; US 4937828 ; WO 8910029

APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
CA 2037219C	February 27, 1991	1991CA-2037219	
EP 444661A	February 28, 1991	1991EP-0102999	
JP06045952A	February 19, 1991	1991JP-0045470	
US 5309449A	February 20, 1991	1991US-0658219	
IT 1241429B	March 1, 1990	1990IT-0067146	
EP 444661B1	February 28, 1991	1991EP-0102999	
DE69120698E	February 28, 1991	1991DE-0620698	
DE69120698E	February 28, 1991	1991EP-0102999	
DE69120698E		EP 444661	Based on

INT-CL (IPC): G06F 11/10; H03M 13/00; H04L 0/00

ABSTRACTED-PUB-NO: EP 444661A
BASIC-ABSTRACT:

A shift register is implemented with flip-flops (FF1-FF4) equal in number to the desired deg. of the remainder polynomial, and two Exclusive-OR gate modulo-2 adders (EO1, EO2) providing the feedback relevant to terms of the first and fourth deg. respectively of the divisor polynomial.

A derivative logic circuit comprises a clock signal inverter (NO), two flip-flops (FF5, FF6) and an AND gate (AN) which is opened by their true and complement outputs respectively.

ADVANTAGE - Easily integrated, allows reset operations to be performed concurrently with loading of first bit of each data block.
ABSTRACTED-PUB-NO:

EP 444661B
EQUIVALENT-ABSTRACTS:

An electronic circuit for generating an error detection code for errors in digital signals (D), organized in data blocks forming the coefficients of a dividend polynomial, to be divided by a suitable divisor polynomial of (n)th degree to obtain a remainder polynomial of (n-1)th degree, whose coefficients (R1,R2,R3,R4) form the error detection code, the circuit consisting of a shift register composed of n flip-flops (FF1,...,FF4) wherein the signal outgoing from the (i-1)th flip-flop (FF1), i being comprised between 0 and n, is sent to a first input of (i)th flip-flop (FF2) directly or through a modulo-2-adder (EO1) according to whether the coefficient to the term of (i)the degree of th divisor polynomial is equal to or different from zero, said adder (EO1) adding to the signal outgoing from the preceding flip-flop (FF1) a feedback signal (10) obtained at the output of the (n)th flip-flop (FF4) which is the MSB flip-flop, either directly, or after adding it to the input data signal (D) in a modulo-2-adder (EO2) which for providing the feedback signal (10) receives as inputs the MSB of the remainder (R4) and said input data signal (D), the feedback signal (10) being supplied to the (i-1) th flip-flop (FF1) and to the other modulo-2 adder (EO1), and all the flip-flops being clocked by a common clock signal (C), characterized in that a second input of the (i)t flip-flop (FF2) accepts the first bit of said data block or a bit at low logic value, according to whether the coefficient of the term of (i)th degree of th divisor polynomial to respectively different from or equal to zero, in correspondence with the clock signal (C) transition relevant to the first bit of the data block, whilst at the subsequent transitions the (i)th flip-flop accepts the signals present at the first input, the first or second inputs being enabled according to the logic level of a signal (L) active in correspondence with the first bit of a data block.

US 5309449A

The electronic system includes: a first flip-flop (FF1) receiving a data signal (D) from a data input, a logic signal (L) common to all of the flip-flops of the shift register, a clock signal (C) common to all of the flip-flops of the shift register, and a feedback signal (10). There is a second flip-flop (FF2) receiving an adder signal (11), the data signal (D), the clock signal (C) and the logic signal (L). At least one further flip-flop (FF3) receives an input from an output of a preceding flip-flop of the shift register, the clock signal (C) and the logic signal (L) and having an output connected to an input of a subsequent flip-clop of the shift register. A final flip-flop (FF4) has an input from a preceding flip-flop of the shift register, the clock signal 1(C) and the logic signal (L).

An adder (EO1) receives an output from the final flip-flop and the data signal (D) and generates the feedback signal (10). Another adder (EO2) receives an output from the first flip-flop and from the feedback signal (10) and generates the adder signal (11).

USE - For generating error detection code in digital signals, organised in data block (D).

CHOSEN-DRAWING: Dwg.1/2 Dwg.1/2 Dwg.1/2

TITLE-TERMS: DIGITAL SIGNAL ERROR CORRECT CODE GENERATOR REMAINING OBTAIN POLYNOMIAL

DIVIDE DIVIDE POLYNOMIAL COEFFICIENT DEFINE BIT SERIAL DATA BLOCK

DERWENT-CLASS: U21 W01

EPI-CODES: U21-A06; W01-A01;

SECONDARY-ACC-NO:
Non-CPI Secondary Accession Numbers: N1991-199447